

İzzettin Furkan Özmen / 2121251024

Computer Programming III Laboratory

Midterm Exam

15/12/2023

## Computer Programming III Laboratory Project Report

### “ Inventory Tracking System”

#### 1. Aim of The Project

The Inventory Tracking System is a comprehensive software solution designed to manage items, users, and various inventory operations. The project utilizes C programming language with a focus on console-based interactions. The key features include inventory management, user authentication, and file-based data storage. This report provides insights into the planning, execution, and testing phases of the Inventory Tracking System.

#### 2. Program Structs and Relationships

##### 2.1) The main purpose of use of struct User:

“**Struct User**” is employed to store user information, enabling functionalities such as user registration, login, and distinguishing between user types (normal user and admin). This structure forms the foundation for user authentication and access control.

##### 2.2 The main purpose of use of struct UserType:

“**Struct UserType**” is utilized to categorize users into different types, defining their roles and permissions within the system. This classification ensures that each user operates within the prescribed boundaries based on their user type.

##### 2.3 The main purpose of use of struct Item:

“**Struct Item**” represents the core structure for storing information about inventory items. It includes details such as item ID, name, price, quantity, and category. Instances of this structure populate the inventory, forming the basis for inventory-related operations.

##### 2.4 The main purpose of use of object InventoryData:

**struct InventoryData** serves as a container for managing the overall inventory system. It encapsulates arrays of items (**struct Item**) and users (**struct User**), along with respective counts.

This structure facilitates efficient data organization and manipulation.

- **inventory**: An array of **struct Item** to store details about each inventory item (e.g., ID, name, price, quantity, category).
- **itemCount**: An integer representing the total number of items in the inventory.
- **users**: An array of **struct User** to store user information (e.g., username, password, userType).
- **userCount**: An integer representing the total number of users in the system.

## 2.5 The main purpose of use of object UserFunctions:

**countUsers**: A function pointer pointing to a function that counts the total number of users. This function takes a filename parameter, indicating the source file for user data.

**countAdmins**: A function pointer pointing to a function that counts the total number of admin users. Similar to **countUsers**, it also takes a filename parameter.

“**Struct UserFunctions**” introduces function pointers for dynamic counting of users and admins, allowing flexibility in handling user-related tasks.

## 3 Rules:

**3.1.** The code includes a function **authenticateUser** for authenticating users based on their username and password. User authentication is used in the login process, providing access to different functionalities based on the user type.

**3.2.** The system differentiates between normal users and admin users based on the user type. Admin users have additional functionalities compared to normal users.

**3.3.** If an item with the same ID exists, it displays an error message, and the item is not added. Similarly, when adding a new item, the code checks if an item with the same name already exists. It compares the new item's name with existing names (case-sensitive) and prevents adding duplicates.

**3.4.** When updating an existing item, the code ensures that the new name is unique among existing items. If an item with the same name (excluding the case) exists, the update is rejected.

**3.5.** When adding or updating an item, the code ensures that the entered values for price and quantity are non-negative. If the input is invalid, the user is prompted to re-enter the correct values. The system ensures that the stock quantity of an item is not allowed to go negative after a sale. This is typically handled by validating the quantity input during updates and sales.

3. **3.6.** The code checks whether the maximum number of items or users has been exceeded before allowing additions. If the maximum limit is reached, it displays an error message.